METHODS AND SYSTEMS FOR PROVIDING DATABASE NODE ACCESS CONTROL FUNCTIONALITY IN A COMMUNICATIONS NETWORK ROUTING NODE

AN APPLICATION FOR

UNITED STATES LETTERS PATENT

By

Paul Andrew Miller
Raleigh, North Carolina

Robby Darren Benedyk
Raleigh, North Carolina

Venkataramaiah Ravishankar
Apex, North Carolina

Peter Joseph Marsico
Carrboro, North Carolina

"Express Mail" mailing number: EJ227696673US
Date of Deposit: March 29, 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231
Lynette M. Bailey
Bailey

Description

METHODS AND SYSTEMS FOR PROVIDING DATABASE NODE ACCESS CONTROL FUNCTIONALITY IN A COMMUNICATIONS NETWORK ROUTING NODE

Priority Application Information

Sub A1

~~This application is a continuation-in-part of U.S. Patent Application Serial No. 09/205,809, filed December 4, 1998 (pending), a continuation-in-part of U.S. Patent Application Serial No. 09/443,712, filed November 19, 1999 (pending), and claims the benefit of U.S. Provisional Patent Application Serial No. 60/127,889, filed April 5, 1999, the disclosures of each of which are incorporated herein by reference in their entirety.~~

Technical Field

The present invention relates to the routing of signaling messages in a communications network, and more particularly to methods and systems for routing signaling messages destined for database network elements.

Background Art

In modern telephony networks, service control points (SCPs) serve as an interface to telephony related databases, such as: call management services databases (CMSDB); line information databases (LIDB); and business services databases (BSDB). These databases are used, at least in part, to facilitate a variety of advanced intelligent network (AIN) related services including: find me service; follow me service; computer security service; call pickup service; store locator service; call waiting service; call block service; calling name delivery service; three way calling service; and 800 number services.

With particular regard to find me service, this service allows calls to be forwarded to another location. The difference between this feature and current call forwarding functionality is the ability to screen unwanted calls from forwarding. Only authorized callers are forwarded to the new location. Similarly, follow me service allows a number to be forwarded on a time schedule. The subscriber determines the time forwarding is to take place when the feature is invoked. Destinations can include both wired and wireless telephones or handsets.

Computer security service allows subscribers to prevent unauthorized callers from accessing a computer or application services. Only callers with the authorized access code or calling from an authorized number can access the services. The SS7 network delivers the calling party number to the destination end office. This number is then checked in a database located with a service control point (SCP), and, if authorized, is allowed to connect with the application. With call pickup service, when a call is placed to a number and is unanswered, the called party can be paged via radio pager. The called party can then dial a code from any telephone at any location and immediately be connected with the waiting caller. With regard to paging type services, manufacturers of such personal communications services (PCS) devices have to date developed two-way pagers that connect a caller with the party being paged. The pager is a two-way transceiver capable of receiving calls (pages) and connecting the caller with the paged party.

Store locator service allows businesses to advertise one number, and have callers automatically transferred to the nearest location based on the caller's telephone number. This allows businesses to advertise nationwide for all locations without special ads that are region specific. The calling party number is matched in a routing database located at an SCP, and the SCP provides the end office with the routing instructions based on the calling party number. With call routing service, businesses can reroute calls during periods of excessively high call volumes or after business hours.

Sub A2

It will be further appreciated that such telephony service databases may also be employed to provide communication service subscribers the flexibility to easily port their service from one communication service provider to another (i.e., number portability or local number portability). The application of such SCP-type database services is not limited to the traditional wired public switched telephone network (PSTN), but is also widely implemented in the wireless telecommunications industry. Typical wireless network communication database applications include; home location registers (HLRs), visitor location registers (VLRs), authentication centers (AuCs), and equipment identification registers (EIRs). In general, SCPs are the network elements that include database systems for providing the services discussed above.

It will also be appreciated that with the continuing convergence of traditional data networks and traditional telecommunication networks, the number and variety of converged or inter-network service related database applications designed to service the needs of combined data-telecommunications subscribers (e.g., presence service databases) will increase dramatically in the future.

Sub A3

With particular regard to traditional SCP network database elements, those skilled in the art of telecommunication network services will appreciate that an SCP is typically comprised of both a front end computer processor system and a back end database system. That is, the SCP front end processor (FEP) system typically does not store or contain the bulk data or information, but instead is the interface to a mainframe or minicomputer system that holds the actual database. Typically, there is a one-to-one correspondence between each FEP and an associated back end computing platform. In a signaling system 7 (SS7) signaling network environment, communication between an SCP front end and other nodes in the SS7 network is accomplished via dedicated SS7 communication links, while communication between the SCP front end and mainframe database back end is typically affected via a TCP/IP connection (or X.25 in older legacy systems). However, it should be noted that even within the telecommunications industry it is not uncommon to hear the term SCP used to describe the combination of front-end processors and mainframe back end database systems.

From an accessibility standpoint, the SS7 network address component of an SCP front end is a point code (PC), while the address component of an application residing on the database back end is referred to as a subsystem number (SSN). A single SCP may contain multiple applications and databases, and as such, there may be multiple subsystem numbers associated with a single SCP point code. Consequently, each SCP must be assigned a unique SS7 network address PC, but may have multiple back end database subsystems provisioned under each unique SS7 network address PC.

Typically, the front end of an SCP located in an SS7 network can perform protocol conversion from SS7 to TCP/IP (or SS7 to X.25 in the case of legacy systems), or it may provide the capability of communicating with the associated back end database directly through the use of primitives. A primitive is an interface that provides access from one level of the protocol to another level. In the case of back end databases, each database is considered to be an application entity, and the protocol used to access and interface to each application entity is known as transaction capabilities application part or TCAP.

Shown in Figure 1 is an example of a prior art telecommunications network, generally indicated by the numeral **100**, that provides AIN-type functionality similar to that described above. Telecommunications network **100** includes an originating end office (EO) or service switching point (SSP) **110**, a signal transfer point (STP) **112**, a first SCP **116**, a second SCP **120**, and a third

SCP **124**. It will be appreciated from Figure 1 that SSP **110** has a network address PC of 3-1-1, STP **112** has a PC of 2-1-1, SCP **116** has a PC of 1-1-1 and a SSN of 20, SCP **120** has a PC of 1-1-2 and a SSN of 20, and SCP **124** has a PC of 1-1-3 and SSN of 20. As further indicated in Figure 1, SSP **110** is coupled to STP **112** via a dedicated SS7 communication link **114**, which is in turn communicatively coupled to each of the three SCP nodes via dedicated SS7 communication links **118**, **122**, and **126**. With regard to the SCP nodes **116**, **120**, and **124**, it will be appreciated from Figure 1 that each overall SCP node is comprised of a number of components or sub-systems. More particularly, SCP **116** generally includes a front end processor (FEP) **128**, which is coupled to a back end database (BED) **130** via a communication link or bus **132**.

Given the above description of network **100**, it will be appreciated by one skilled in the art of telecommunication signaling operations that if, for instance, Calling Name (CNAM) service is requested by a subscriber that is serviced by SSP **110**, then SSP **110** will be required to formulate and send a CNAM query-type SS7 signaling message to STP **112** via the dedicated SS7 communication link **114**. If it is also assumed that a database application corresponding to SSN 20 of SCP **116** is provisioned to provide CNAM-type information, then CNAM query message will either be addressed directly to the PC & SSN of SCP **116** (i.e., PC: 1-1-1, SSN: 20), or the CNAM query message will be addressed so as to request a final destination address translation at the STP **112** (i.e., through global title translation). For purposes of illustration, it is assumed that global title translation service is not required and, consequently, that the CNAM query message is addressed directly to SCP **116** (i.e., PC: 1-1-1, SSN: 20). As such, the CNAM query message is received by STP **112** and subsequently routed over communication link **118** to FEP **128**. FEP **128** in turn receives the CNAM query message, processes the message, and facilitates access to the CNAM data stored in the BED **130**. Ultimately, a CNAM reply message addressed to SSP **110** (PC: 3-1-1) is formulated and transmitted back to STP **112**, which in turn routes the message to SSP **110**.

As described above, each complete SCP unit is equipped with a front end processor that is responsible for managing the unit's associated database resources. Such management functions include: protocol conversion, message parsing, administration of inbound queries and outbound responses, load sharing, etc. Each front end processor is integral with the SCP unit and consequently, there is a one-to-one relationship that exists between front end processors and SCP units. Front end processors are expensive, and what is needed is a way to reduce the overall cost of SCP units by allowing one front end processor to drive multiple SCP units.

Therefore, what is needed is a system and method of incorporating SCP front end processing functionality within a communications network routing node such that multiple SCP back ends can be serviced by the single routing node. Furthermore, the SS7 signaling links typically employed to connect to SCP units are capital intensive and expensive to maintain. Consequently, a method of connecting to SCP units that does not require dedicated, expensive SS7 signaling links is also needed.

Disclosure of the Invention

According to one aspect, the present invention includes a communications network element that is capable of generally routing messages and also performing load sharing, protocol conversion, and other services that have traditionally been provided by SCP front end processing (FEP) modules. The FEP routing node includes a communication module or modules capable of transmitting and receiving data packets over both SS7 and IP networks. A message discrimination process examines incoming data packets and subsequently directs certain packets to a database access control process that administers database lookup, protocol translation, and other FEP related processing services.

The functions for providing database access control are described herein as modules or processes. It is understood that these modules or processes may be implemented as computer-executable instructions embodied in a computer-readable medium. Alternatively, the modules or processes described herein may be implemented entirely in hardware. In yet another alternative embodiment, the modules or processes described herein may be implemented as a combination of hardware and software.

The processes and modules for providing database access control are described below as being associated with cards or subsystems within a routing node. It is understood that these cards or subsystems include hardware for storing and executing the processes and modules. For example, each card or subsystems described below may include one or more microprocessors, such as a Pentium® processor available from Intel Corporation, and associated memory.

Accordingly, it is an object of the present invention to provide a routing node that offers centralized front end processing functionality to multiple SCP back ends.

It is yet another object of the present invention to provide a routing node that facilitates load sharing functionality among multiple SCP back ends.

It is yet another object of the present invention to provide a routing node that facilitates message protocol translation.

It is yet another object of the present invention to provide a method of eliminating the need for SS7 network point codes associated with SCP nodes.

It is yet another object of the present invention to provide a method of creating a virtual SCP that is comprised of multiple SCP back end databases, where the virtual SCP is assigned a single SS7 network point code.

It is yet another object of the present invention to provide a method of mapping multiple SCP nodes to a single SS7 network point code.

It is yet another object of the present invention to provide a method of allowing all messages requiring SCP service to be addressed to an SS7 point code that is the same as the SS7 point code of a router of the present invention.

It is yet another object of the present invention to provide a method of allowing messages to be routed based on the ownership of a database or SCP node.

Some of the objects of the invention having been stated hereinabove, other objects will become evident as the description proceeds, when taken in connection with the accompanying drawings as best described hereinbelow.

Brief Description of the Drawings

Embodiments of the invention will now be described with reference to the accompanying drawings, of which:

Figure 1 is a network diagram illustrating a prior art telecommunications network that employs a signal transfer point (STP) node and multiple service control point (SCP) nodes;

Figure 2 is a schematic diagram of an STP switching node;

Figure 3 is a schematic and message flow diagram of a system architecture according to a preferred embodiment of a packet routing node of the present invention, generally indicating message flow associated with an incoming SCP query packet;

Figure 4 is a table that illustrates a sample database access control (DAC) database structure and data used in a preferred embodiment of a packet routing node of the present invention;

Figure 5 is a schematic diagram of a system architecture according to another embodiment of a packet routing node of the present invention, generally illustrating an externally mounted DAC process;

Figure 6 is a flow chart diagram illustrating integrated front end processor (FEP) processing of a SCP query message according to an embodiment of a packet routing node of the present invention;

Figure 7 is a schematic and message flow diagram of a system architecture according to a preferred embodiment of a packet routing node of the present invention, generally indicating message flow associated with an incoming network status packet;

Figure 8 is a network diagram illustrating an embodiment of the present invention that includes a packet routing node with FEP functionality and multiple SCP nodes;

Figure 9 is a network diagram illustrating another embodiment of the present invention where multiple SCP nodes each are assigned the same network address point code;

Figure 10 is a network diagram illustrating another embodiment of the present invention where multiple virtual service control point (vSCP) nodes each are assigned the same network address point code; and

Figure 11 is a network diagram illustrating another embodiment of the present invention where multiple vSCP nodes each are assigned the same network address point code as a packet router node.

Detailed Description of the Invention

Disclosed herein are several embodiments of the present invention, all of which include a network element that performs functions similar to that of a traditional telecommunications network packet routing switch, such as a signal transfer point (STP). Each of the embodiments described and discussed below, employs an internal architecture similar to that of high performance STP and signaling gateway (SG) products which are marketed by Tekelec, Inc. of Calabasas, California as the Eagle® STP and IP^7 Secure Gateway^{tm}, respectively. A block diagram that generally illustrates the base internal architecture of the IP^7 Secure Gateway^{tm} product is shown in Figure 2. A detailed description of the Eagle® STP may be found in the *Eagle® Feature Guide* PN/910-1225-01, Rev. B, January 1998, published by Tekelec, the disclosure of which is incorporated herein by reference in its entirety. Similarly, a detailed description of the IP^7 Secure Gateway^{tm} may be found in Tekelec publication PN/909-0767-01, Rev B, August 1999, titled *Feature Notice IP^7 Secure Gateway^{tm} Release 1.0*, the disclosure of which is incorporated by reference in its entirety. The specific functional components of an IP^7 Secure Gateway^{tm} for transmitting and receiving TCAP messages over an Internet protocol (IP) network are described in above-referenced, co-pending U.S. Patent Application Serial No. 09/205,809. As described in the above referenced *Eagle® Feature Guide*, an Eagle® STP **250** includes the following subsystems: a maintenance and administration subsystem (MAS) **252**, a communication subsystem **254** and an application subsystem **256**. The MAS

252 provides maintenance communications, program load, user interface, alarm processing and system disks. The communication subsystem **254** includes an interprocessor message transport (IMT) bus that is the main communication bus among all subsystems in the Eagle® STP **250**. This high-speed communications system functions as two 125 Mbps counter-rotating serial buses.

The application subsystem **256** includes application cards that are capable of communicating with the other cards through the IMT buses. Numerous types of application cards can be incorporated into STP **250**, including but not limited to: al link interface module (LIM) **258** that provides SS7 links and X.25 links, a database communication module (DCM) **260** that provides an IP interface using transmission control protocol (TCP), and an application service module (ASM) **262** that provides global title translation, gateway screening and other services. A translation service module (TSM) **264** may also be provided to support triggered local number portability service. Once again, a detailed description of the Eagle® STP is provided in the above-cited *Eagle® Feature Guide* and need not be described in detail herein. It should also be appreciated that, in addition to conventional SS7 LIM cards, a database communication module (DCM) can be employed in a similar manner to provide for the transport of IP encapsulated SS7 messages over an IP network, as described in the above referenced *Feature Notice IP*^{7} *Secure Gateway*^{tm} *Release 1.0* publication. With particular regard to the TSM triggered LNP services module mentioned above, a detailed description of the Tekelec triggered LNP solution may be found in the *Feature Guide LNP LSMS* PN/910-1598-01, Rev. A, January 1998, published by Tekelec, the disclosure of which is incorporated herein by reference in its entirety. Furthermore, systems and methods for providing triggerless LNP functionality within a network routing node are described in commonly-assigned, co-pending U.S. Patent Application Serial No. 09/503,541, the disclosure of which is incorporated herein by reference in its entirety.

Integrated DAC Database Embodiment

Shown in Figure 3 is a front end processing (FEP) packet routing node of the present invention that is generally indicated by the numeral **500**. It will be appreciated that FEP routing node **500** is communicatively coupled to an EO or SSP **110** via an SS7 signaling link **212**, and to an IP data network **216** via an IP connection **218**. As further illustrated in Figure 3, FEP packet routing node **500** includes a high speed interprocessor message transport (IMT) communications bus **504**. Communicatively coupled to IMT bus **504** are a number of distributed processing modules or cards including: a pair of maintenance and administration subsystem processors (MASPs) **506**; an SS7 capable link interface module (LIM) **502**; an IP capable database communication module (DCM) **510**; and a database access control module (DACM) **508**. These modules are physically connected to the IMT bus **504** such that signaling and other type messages may be routed internally between all active cards or modules. For simplicity of illustration, only a single LIM **502**, DCM **510** and

DACM **508** are included in Figure 3. However, it should be appreciated that the distributed, multi-processor architecture of the FEP routing node **500** facilitates the deployment of multiple LIM, DCM, DACM and other cards, all of which could be simultaneously connected to and communicating via IMT bus **504**.

MASP pair **506** implement the maintenance and administration subsystem functions described above. As the MASP pair **506** are not particularly relevant to a discussion of the flexible routing attributes of the present invention, a detailed discussion of their function is not provided herein. For a comprehensive discussion of additional MASP operations and functionality, the above-referenced Tekelec publications can be consulted.

~~Focusing now on LIM card functionality, it will be appreciated that LIM **502** is comprised of a number of sub-component processes including, but not limited to; SS7 MTP level 1 and 2 processes **512**, an I/O buffer or queue **514**, an SS7 MTP level 3 HMDC process **516**, and an HMDT process **518**. MTP level 1 and 2 processes **512** provide the facilities necessary to send and receive digital data over a particular physical media / physical interface, as well as to provide error detection / correction and sequenced delivery of all SS7 message packets. I/O queue **514** provides for temporary buffering of incoming and outgoing signaling message packets. MTP level 3 HMDC process **516** receives signaling messages from the lower processing layers and performs a discrimination function, effectively determining whether an incoming SS7 message packet requires internal processing or is simply to be through switched. HMDT process **518** handles internal routing of SS7 message packets~~

Sub A4

that require additional processing prior to final routing. Once again, it should be appreciated that a LIM card may contain more functional processes than those described above. The above discussion is limited to LIM functionality associated with the basic processing of in-bound signaling messages.

DCM **510**, shown in Figure 3, generally includes an I/O buffer or queue **540** and an IP level 1 & 2 process **542**. It will be appreciated that outgoing message packets routed through the DCM **510** will be transmitted out of the FEP routing node **500** and into IP network **216** via IP communication link **218**. As the SS7 and IP communication protocols are not inherently compatible, all SS7 message packets that are to be sent into the IP network **216** are first encapsulated within a TCP/IP routing envelope prior to transmission. This IP encapsulation is performed on the DCM **510** by the IP level 1 & 2 process **542**. Preferred packet formats for encapsulating various types of SS7 messages in IP packets are described in Internet Engineering Task Force (IETF) INTERNET DRAFT entitled “Transport Adapter Layer Interface”, May 28, 1999, the disclosure of which is incorporated herein by reference in its entirety. Furthermore, a Tekelec Transport Adapter Layer Interface (TALI^{tm}) is described in commonly-assigned, co-pending U.S. Provisional Patent Application Serial No. 60/137,988, the disclosure of which is incorporated herein by reference in its entirety.

Once again, the description of LIM and DCM sub-components provided above is limited to those sub-components that are relevant to the sample implementation scenarios illustrated herein. For a comprehensive discussion of additional LIM and DCM operations and functionality, the above-referenced Tekelec publications can be consulted.

With regard to DACM card **508**, it will be appreciated from Figure 3 that DACM generally includes the database and control processes necessary to achieve the front end processing (FEP) functionality of the present invention. The DACM **508** shown in Figure 3 is comprised, in part, of a signaling connection control part (SCCP) subsystem **520**, an SCCP controller, known as a signaling connection routing controller (SCRC) process **522**, and a database access control (DAC) process **524**. SCCP subsystem **520** is responsible for receiving and preliminary processing of incoming SCCP protocol message packets. The SCRC process **522** is responsible for discrimination of signaling messages at the SCCP level, and for distributing the signaling messages to a higher processing level when appropriate. In the configuration shown in Figure 3, the next highest processing level is represented by the DAC process **524**.

DAC process **524** includes a DAC database **526** and a DAC protocol translation process **528**, as indicated in Figure 3. DAC process **524** is generally responsible for examining properties or characteristics of an incoming message and determining what, if any, processing is required. Such incoming message properties or characteristics might include, but are not limited to: the origination point code (OPC); destination point code (DPC); subsystem (SSN); and translation type (TT). DAC process **524** is also responsible for monitoring and storing information related to the operating status of network database or SCP nodes which have been provisioned for FEP servicing by the FEP routing node

500. Such operating status information is also stored in DAC database 526 and might include, but is not limited to: node In Service / node Out Of Service indicators; overall node congestion indicators; and link specific congestion indicators. In addition to operating status type information, DAC database 526 can also contain information related to database or SCP node ownership. Consequently, message routing decisions can be based, at least in part, upon database or SCP node ownership. Along with such operating status and ownership information, the DAC database 526 maintains a set of SS7 to IP routing address translation instructions, all of which are generally illustrated in Figure 4. It will be further appreciated that DAC protocol translation process 528 is provisioned to translate an incoming database query or response message into any of a variety of provisioned database protocols (e.g., structured query language (SQL), open database connectivity (ODBC), etc.) depending upon the protocol dictated by a particular SCP or database node. Once DAC processing is complete, the resulting message is passed to an HMRT process 532 for internal routing to the appropriate outbound LIM or DCM module.

It will be appreciated from Figure 3 that DACM 508 is in communication with and serviced by an Operations Administration and Maintenance (OAM) system 550. In general, an OAM system provides a mechanism whereby network routing address information contained within the DAC process 524 can be externally provisioned or dynamically updated. As the interaction between FEP routing node 500 and OAM 550 is not particularly relevant to the present invention, a detailed discussion of such OAM system functionality will not be presented herein. It should suffice to state that the OAM **550** maintains the routing database component of the DAC process **524** with the most current network routing address information available at any given time.

In the embodiment shown in Figure 3, the DAC process **524** resides in one or more blocks of high speed random access memory (RAM) that are located on DACM card **508**. However, it will be appreciated by those skilled in the art of high-performance computing systems that such a software process and any databases associated therewith could be configured such that some or all of the information is stored on a high density, fast access physical storage media such as magnetic or optical discs.

As indicated in Figure 4, the DAC database component **526** is comprised of a series of entries or records, with each record containing a number of data fields including, but not limited to: a point code (PC) field; a subsystem (SSN) field; an IP host name field; an IP port field; a database or SCP node protocol field; a service or translation type (TT) field; a node status field; a node congestion field; and an owner field.

Once again, DACM **508** also contains HMRT process **532** that is responsible for the routing of message packets once DAC processing has been completed. That is, the HMRT process **532** determines to which DCM or LIM card a message packet should be internally routed for subsequent outbound transmission into the communication network.

Shown in Figure 5 is another embodiment of the FEP routing node of the present invention, generally indicated by the numeral **600**. FEP routing node **600** is identical in overall function to the FEP routing node embodiment illustrated in Figure 3 and described above. In most respects, the form of FEP routing node **600** is identical to the FEP routing node **500** shown in Figure 3. That is, FEP packet router node **600** generally includes a high speed interprocessor message transport (IMT) communications bus **504**, and a number of distributed processing modules or cards including; a pair of maintenance and administration subsystem processors (MASPs) **506**, an SS7 capable link interface module (LIM) **502**, an IP capable database communication module (DCM) **510**. Once again, it will be appreciated that these modules are physically connected to IMT bus **504** such that signaling and other type messages may be routed internally between all active cards or modules and that, for simplicity of illustration, only a single LIM **502** and DCM **510** are depicted. However, as with node **500**, it should be appreciated that the distributed, multi-processor architecture of FEP routing node **600** also facilitates the deployment of multiple LIM, DCM and other cards, all of which could be simultaneously connected to and communicating via IMT bus **504**.

In the case of FEP routing node **600**, it will be appreciated from Figure 5 that the functionality of DACM card **508**, as described above, is now provided by a DACM card **610** in combination with an external database access control (DAC) server **620**. Once again, it will be appreciated the combination of DACM card **610** and DAC server **620** includes the database and control processes necessary to achieve the front end processing (FEP) functionality of the present invention. The DACM card **610** shown in Figure 5 includes a signaling connection control part (SCCP) subsystem **520**, a description of which was provided above. Also, as with the previously discussed embodiment, DACM card **610** includes an SCCP controller known as a signaling connection routing controller (SCRC) process **522**. However, unlike the previous embodiment described above, DACM **610** employs a high-speed Ethernet controller (EC) process **612**. Once again, as described above, the SCCP subsystem **520** is responsible for receiving and preliminary processing of incoming SCCP protocol message packets, while the SCRC process **522** is responsible for discrimination and subsequent distribution of signaling messages at the SCCP level. In the case of DACM card **610**, messages that satisfy the SCRC discrimination criteria are distributed or directed to the high-speed Ethernet controller process **612**. EC process **612** is in turn responsible for controlling the process of communicating messages, via an Ethernet connection **630**, to and from the associated DAC server **620**. More particularly, DAC server **618** includes a corresponding high-speed Ethernet controller process **622** that serves as the communications interface between DACM card **610** and an on-board DAC process **524**. Once again, it will be appreciated that DAC process **524** generally includes a DAC database process **526** and a DAC protocol translation process **528**, and is responsible for determining whether front end processing service is to be provided by the FEP routing node. DAC process **524** is also responsible for monitoring and storing information related to the operating status of provisioned network database or SCP nodes. As discussed previously, in addition to operating status type information, DAC process also contains information related to database or SCP node ownership. Consequently, message routing decisions can be based, at least in part, upon database or SCP node ownership. Along with such operating status and ownership information, the DAC database process **526** maintains a set of routing address translation instructions, which are generally illustrated in Figure 4. It will be further appreciated that DAC protocol translation process **528** is provisioned to translate incoming database query and response messages into any of a variety of database query protocols (e.g., SQL, ODBC, etc.) depending upon the database protocol dictated by a particular destination SCP or database node. Once DAC processing is complete, the resulting message is passed to an HMRT process **532** for internal routing to the appropriate outbound LIM or DCM module.

Once again, it will be appreciated from Figure 5 that DAC server **620** is in communication with and serviced by an operations administration and maintenance (OAM) system **528**, in much the same manner as that described above for DACM **508** in Figure 3.

In the embodiment shown in Figure 5, the DAC process **524** resides in one or more blocks of high speed random access memory (RAM) that are located within the DAC server **620**. However, it will be appreciated by those skilled in the art of high-performance computing systems that such a software process and any databases associated therewith could be configured such that some or all of the information is stored on a high density, fast access physical storage media such as magnetic or optical discs.

Once again, as indicated in Figure 4, the DAC database component **526** of DAC server **620** is comprised of a series of entries or records, with each record containing a number of data fields including, but not limited to: a point code (PC) field, a subsystem (SSN) field, an IP host name field, an IP port field, a database or SCP node protocol field, a service or translation type (TT) field, a node status field, a node congestion field, and an owner field.

DAC Query Transaction Processing

For purposes of illustration, the path of a typical SS7 TCAP query message requiring FEP routing node service is traced, in Figure 3, from reception at the FEP routing node **500** by the inbound LIM **502**, through processing by DACM card **508**, and on to the outbound DCM **510**. A detailed flow chart of FEP related query message processing steps is presented in Figure 6, and may be used in conjunction with the schematic diagram shown in Figure 3 to better understand FEP servicing methodology.

Beginning with step **ST1**, an incoming TCAP query message is received at the inbound LIM module **502**. In step **ST2**, the incoming TCAP query message is received and processed by the MTP Level 1 and 2 process **512**. With MTP Level 1 and 2 processing complete, the signaling message packet is temporarily buffered in the I/O queue **514** before being passed up the stack to the MTP Level 3 HMDC process **516**, where SCCP type discrimination processing is performed. In the example shown in Figure 2, HMDC process

516 examines the message packet and determines that the DPC of the packet is the PC (2-1-1) of the FEP routing node **500** (**ST3**). Consequently, further processing of the SCCP MSU within the FEP routing node is assumed to be necessary, and the packet is passed to the HMDT process **518**.

In this particular example, it is assumed that the FEP routing node **500** is provisioned to respond to query messages that are addressed to the true or capability point code of the FEP routing node **500**. However, as mentioned previously, FEP routing node **500** could easily be provisioned to provide FEP-type processing in response to many point codes other than that of the FEP routing node. While it may prove to be advantageous for service providers to implement an FEP routing node of the present invention in a communications network in a manner such that all query messages requiring FEP-type processing are addressed to the same PC as that of the FEP routing node, it is not essential to the operation of the present invention.

The HMDT process **518** examines the service indicator (SI) field of the incoming TCAP MSU, which indicates that the message is of an SCCP type. As such, HMDT process **518** places the incoming MSU on high speed IMT bus **504** for transport to DACM **508** and subsequent FEP servicing (**ST5**).

The internally routed TCAP MSU is received by the DACM resident SCCP process **520**, and subsequently examined by SCRC process **522** that is resident on DACM card **508**. Upon successful verification, the TCAP MSU is directed to DAC application **524** for further processing. DAC application processing begins with a general determination of incoming message type.

Following the determination that the message is a TCAP-type query message, DAC process **524** proceeds with verification of the pointers and field lengths associated with the TCAP message. Given that the message is a TCAP-type query message, a lookup is performed in DAC database **526** based on PC and SSN information contained within the message (**ST6**). Referring again to Figure 4, it will be appreciated that in the case of an incoming TCAP query message addressed to PC: 2-1-1 and SSN: 50, the DAC database lookup would return two matching records. DAC process **524** then examines both returned routing translation records and makes a final routing decision based on a pre-defined set of selection rules or conditions.

Once again, as indicated in Figure 4, it will be appreciated that one of the two returned routing translation records contains a status field value which indicates that the SCP or database node with which it is associated is currently out of service (OOS). Clearly, routing the TCAP query message to an SCP or database node that is OOS would not be desirable. Thus, in this particular example, DAC process **524** opts to route the incoming TCAP query message to the SCP or database node with an IP address of 10.20.30.40: port 5230 (**ST7**). As such, routing label information within the message packet is modified to reflect this change of destination routing address. In step **ST8,** using information returned by the DAC database lookup that identifies the protocol employed by the database residing at IP address 10.20.30.40: port 5230, DAC process **524** next directs the TCAP query to the DAC protocol translation process **528**. In this case, DAC protocol translation process **528** uses the database query information contained within the TCAP message to construct an equivalent SQL query statement. This new SQL query statement is then substituted for the original database query content of the incoming TCAP message.

With message routing address translation and query protocol translation processing complete, the modified query message is next passed to HMRT process **532** for internal routing to the appropriate DCM card (**ST9**). Consequently, the modified message packet is internally routed across the IMT bus **504** to DCM **510**, where it is generally received by the I/O queue process **540**. Eventually, the modified message packet is passed from the I/O queue process **540** on to the IP Level 2 and Level 1 process **542** where properly formatted IP routing label information is applied to the packet prior to transmission into the associated IP network **216** (**ST10**). Following successful IP Level 1 and 2 processing, the message packet is transmitted into the IP network **216** and generally towards the destination SCP or database node as identified by the previous FEP processing (**ST11**).

It will also be appreciated that the processing of an incoming TCAP query message is performed in a very similar manner for the embodiment of the FEP routing node **600** shown in Figure 5. In the case of the configuration contemplated in Figure 5, messages arriving at the DACM card **610** are simply passed to an external DAC server **620** via a high-speed Ethernet connection **630** prior to DAC database lookup and protocol translation processing. In all other respects, processing of an incoming TCAP query message similar to that described above, would be identical in the FEP routing node **600**.

Although the example presented above relates specifically to the reception and subsequent processing of a TCAP query message bound for an SCP or database node, it will be appreciated by those skilled in the art, that the FEP processing node of the present invention can easily be provisioned to intercept and process subsequent response messages generated by such SCP or database nodes. Such response message processing could include database protocol translation, so as to translate the database response statement into a protocol that is usable by the network element that originated or initiated the query transaction.

DAC Network Management Message Processing

As discussed briefly above, one aspect of FEP processing includes routing address translation that is based, at least in part, on the congestion and overall operational status of potential destination SCP or database nodes. Consequently, the FEP routing node, and more specifically DAC process **524**, must be capable of acquiring and maintaining accurate information related to the status of the SCP or database nodes that are provisioned to have FEP service provided by the FEP routing node.

Given such functional requirements, it will be appreciated that Figure 7 generally illustrates the receipt and subsequent internal processing of a typical network management message received from an SCP or database node residing in or connected to IP network **216**. More particularly, FEP routing node **500** is shown receiving a network management message associated with or sent by an SCP or database node that is provisioned to have FEP service provided by the FEP routing node **500**. The network management message includes information related to the operational status of the related node or the communication pathway(s) that form the effective communication link between the related node and the FEP routing node **500**. Examples of such status information might include, but are limited to; in service / out of service indicators, node congestion indicators, and link congestion indicators.

As indicated in Figure 7, the network management message is received by DCM card **510** and subsequently TCP encapsulated and processed by the IP Level 1 & 2 process **542**. With IP Level 1 and 2 processing complete, the message is passed to and temporarily buffered in I/O queue process **540** before being directed on to HMDC process **538**. HMDC process **538** examines the incoming message packet and determines that the message contains information that is required by one or more DACM cards. Consequently, HMDC process **538** passes the message packet to HMDT process **536** for internal routing to the appropriate DACM card(s). In the example implementation shown in Figure 7, HMDT process **536** internally routes the message packet via IMT bus **504** to the only provisioned DACM card in the system, DACM **508**. It will be appreciated that if multiple DACM cards were simultaneously provisioned in the FEP routing node, HMDT process **536** could direct multiple copies of the network management message packet to each of the provisioned DACM cards connected to IMT bus **504**.

Once the network management message packet is received by DACM card **508**, the message is generally verified and processed by the SCCP and SCRC processes **520** and **522**, respectively. The verified and processed network management message is then passed to the DAC process **524**. DAC process **524** examines the message packet, extracts the necessary node status information, and updates the appropriate records in the DAC database **526**.

Thus, by continuously monitoring and processing network management-type messages generated by the SCP and database nodes provisioned for FEP service by the FEP routing node of the present invention, routing translation data utilized by the FEP routing node to make routing decisions can be maintained in an accurate and useful state.

Once again, it will be appreciated that the FEP routing node configuration shown in Figure 7 includes all DAC related processing modules onboard the DACM card **508**. However, processing of network management messages would be similar in the case of FEP routing node **600**, that is generally illustrated in Figure 5. In the case of the configuration contemplated in Figure 5, messages arriving at the DACM card **610** are simply passed to an external DAC server **620** via a high-speed Ethernet connection **630** prior to updating of the DAC database **526**. In all other respects, processing of an incoming network management message similar to that described above, would be identical in the FEP routing node **500**.

FEP Routing Node Network Implementations

Shown in Figures 8 – 11 are several examples of practical network implementations of an FEP routing node of the present invention. It will be appreciated that the particular embodiment of the FEP routing node chosen to illustrate these sample implementations is the same as that shown in Figure 3 and described in detail above. However, other embodiments of the FEP routing node of the present invention could just as effectively be employed in these sample implementations, including the embodiment illustrated in Figure 5.

Shown in Figure 8 is a typical telecommunications network, generally indicated by the numeral **200**. Telecommunications network **200** includes an end office or SSP **110**, an FEP routing node **500**, an IP data network **216**, and three SCP nodes **220**, **224**, and **228**. It will be appreciated that SSP **110** is communicatively coupled to FEP routing node **500** via an SS7 communication link **212**. However, as mentioned previously, it is not essential that the link **212** be an SS7-type communication link. Such a link could be an IP link carrying encapsulated SS7 signaling messages. In any event, FEP routing node is in turn communicatively coupled to IP network **216** via an IP link or connection **218**. Also connected to IP network **216** via IP connection **222** is SCP **220**. In a similar manner, SCP **224** and SCP **228** are connected to IP network **216** via IP connections **226** and **230**, respectively.

Sub A5

The network configuration shown in Figure 8 is one of the more simple implementations of an FEP routing node of the present invention. When compared with the prior art network configuration illustrated in Figure 1, it will be appreciated that the inclusion of FEP routing node **500** in Figure 8 allows each of the provisioned SCPs 220, **224**, and **228** to eliminate dedicated, internal FEP modules. Consequently, SCP **220** simply includes a computing platform **232** that serves a database back end processor. In a similar manner, SCP **224** and SCP **228** also include back end database computing platforms. Once again, it should be noted that each SCP 220, **224**, and **228** is not required to implement a separate FEP module.

In the example shown in Figure 8, all of the SCP nodes are identified by a unique SS7 point code, each of which is different from the point code assigned to the FEP routing node (2-1-1). More particularly, SCP **220** is assigned a point code of 1-1-1, while SCP **224** has a point code of 1-1-2, and SCP **228** has a point code of 1-1-3. It will also be appreciated that, as each of these SCP nodes is connected to the FEP routing node **500** via the IP data network **216**, each SCP also has a uniquely assigned IP address in the form of a host name and port.

As such, when FEP routing node **500** receives a TCAP query message from SSP **110** that is destined for the SCP with point code of 1-1-1 and SSN of 20, FEP routing node might simply perform database protocol translation-type processing on the message, encapsulate the modified message in a properly addressed IP packet, and route the modified message on to SCP **220**. Any other processing that was previously performed by FEP **128** shown in Figure 1 could also be performed by the FEP routing node **500**.

One of the great advantages of even such a simple implementation of a FEP routing node of the present invention is apparent upon closer examination of Figure 8. This advantage being that a single Front End Processing module, properly integrated within a routing node of the present invention, can accommodate all of the FEP tasks associated with each of the SCPs **220**, **224**, and **228**. Such an architecture presents SCP node owners with a major cost savings, which can ultimately be passed on to the end consumer.

Illustrated in Figure 9 is another typical telecommunications network, generally indicated by the numeral **300**. As with the previously described telecommunications network **200**, network **300** also includes an End Office or SSP **110**, an FEP routing node **500**, an IP data network **216**, and three SCP nodes **220**, **224**, and **228**. It will be appreciated that SSP **110** is communicatively coupled to FEP routing node **500** via an SS7 communication link **212**. FEP routing node **500** is in turn communicatively coupled to IP network **216** via an IP link or connection **218**. Also connected to IP network **216** via IP connections **222, 226** and **230** are SCPs **220**, **224**, and **228**, respectively.

Once again, when compared with the prior art network configuration illustrated in Figure 1, it will be appreciated that the inclusion of FEP routing node **500** in Figure 9 allows each of the provisioned SCPs **220**, **224**, and **228** to eliminate dedicated, internal FEP modules. Consequently, SCP **220** simply includes a computing platform **232** that serves a database back end processor. In a similar manner, SCP **224** and SCP **228** also include back end database computing platforms.

However, it will be appreciated that the network configuration shown in Figure 9 is a slightly more complex implementation of an FEP routing node than that shown in Figure 8. In the example shown in Figure 9, all of the SCP nodes are identified by the same SS7 point code (1-1-1), which is different from the point code assigned to the FEP routing node (2-1-1) and all of the SCPs are provisioned with the same subsystem, SSN 20. However, as each of these SCP nodes is connected to the FEP routing node **500** via the IP data network **216**, each individual SCP will still have a unique assigned IP address in the form of a Host name and Port.

As such, when FEP routing node **500** receives a TCAP query message from SSP **110** that is destined for the SCP with point code of 1-1-1 and SSN of 20, FEP routing node might make a decision to route based on the operational status or congestion status of the three SCPs **220**, **224** and **228**. Such load shedding or load sharing among the three similarly provisioned SCPs will ultimately lead to superior overall network performance. The FEP routing node **500** might further perform database protocol translation-type processing on the message, encapsulate the modified message in a properly addressed IP packet, and route the modified message on to the selected SCP. Once again, any additional processing that was previously performed by FEP **128** shown in Figure 1 could also be performed by the FEP routing node **500**.

Illustrated in Figure 10 is yet another typical telecommunications network, generally indicated by the numeral **400**. As with the previously described telecommunications network **200**, network **400** also includes an end office or SSP **110**, an FEP routing node **500**, an IP data network **216**, and three SCP nodes **410**, **420**, and **430**. It will be appreciated that SSP **110** is communicatively coupled to FEP routing node **500** via an SS7 communication link **212**. FEP routing node **500** is in turn communicatively coupled to IP network **216** via an IP link or connection **218**. Also connected to IP network **216** via the plurality of IP connections **414**, **424** and **434** are SCPs **410**, **420**, and **430**, respectively. In this sample implementation, each SCP is comprised of multiple back end database processors which effectively form a series of virtual SCP (vSCP) nodes. More particularly, SCP **410** is comprised of a series of back end database processors generally indicated by the numeral **412**, all of which are represented by a point code of 1-1-1 and subsystem of 20. In this example, it is assumed that SSN 20 of SCP **410** provides LIDB-type database information. SCP **420** is comprised of a series of back end database processors generally indicated by the numeral **422**, all of which are represented by a point code of 1-1-1 and subsystem of 30. In this example, it is assumed that SSN 30 of SCP **420** provides 800 number-type database information. SCP **430** is comprised of a series of back end database processors generally indicated by the numeral **432**, all of which are represented by a point code of 1-1-1 and subsystem of 50. In this example, it is assumed that SSN 50 of SCP **410** provides CNAM-type database information.

Once again, when compared with the prior art network configuration illustrated in Figure 1, it will be appreciated that the inclusion of FEP routing node **500** in Figure 10 allows each of the provisioned SCPs **410**, **420**, and **430** to eliminate dedicated, internal FEP modules that would have otherwise have been associated with each back end database processor.

It will be appreciated that the network configuration shown in Figure 10 is a still slightly more complex implementation of an FEP routing node even than that shown in Figure 9. In the example shown in Figure 10, all of the SCP nodes are identified by the same SS7 point code (1-1-1), which is different from the point code assigned to the FEP routing node (2-1-1) yet each SCP node has a different subsystem provisioned for service. Once again, as each of the back end processors that comprise these SCP nodes is connected to the FEP routing node **500** via the IP data network **216**, each individual back end processor of each SCP will still have a unique assigned IP address in the form of a host name and port.

As such, when FEP routing node **500** receives a TCAP query message from SSP **110** that is destined for the SCP with point code of 1-1-1 and SSN of 50, FEP routing node might make a decision to route based on the operational status or congestion status of the multiple back end processors **432** that are associated with SCP **430**. Such load shedding or load sharing among multiple, similarly provisioned back end processors will ultimately lead to superior overall network performance. The FEP routing node **500** might further perform database protocol translation-type processing on the message, encapsulate the modified message in a properly addressed IP packet, and route the modified message on to the selected SCP back end processor. Once again, any additional processing that was previously performed by FEP **128** shown in Figure 1 could also be performed by the FEP routing node **500**.

Illustrated in Figure 11 is still another typical telecommunications network, generally indicated by the numeral **450**. As with the previously described telecommunications network **200**, network **450** also includes an End Office or SSP **110**, an FEP routing node **500**, an IP data network **216**, and three SCP nodes **410**, **420**, and **430**. It will be appreciated that SSP **110** is communicatively coupled to FEP routing node **500** via an SS7 communication link **212**. FEP routing node **500** is in turn communicatively coupled to IP network **216** via an IP link or connection **218**. Also connected to IP network **216** via the plurality of IP connections **414**, **424** and **434** are SCPs **410**, **420**, and **430**, respectively. Once again, in this sample implementation, each SCP is comprised of multiple back end database processors which effectively form a series of virtual SCP (vSCP) nodes. More particularly, SCP **410** is comprised of a series of back end database processors generally indicated by the numeral **412**, all of which are represented by a point code of 2-1-1 and subsystem of 20. In this example, it is assumed that SSN 20 of SCP **410** provides LIDB-type database information. SCP **420** is comprised of a series of back end database processors generally indicated by the numeral **422**, all of which are represented by a point code of 2-1-1 and subsystem of 30. In this example, it is assumed that SSN 30 of SCP **420** provides 800 number-type database information. SCP **430** is comprised of a series of back end database processors generally indicated by the numeral **432**, all of which are represented by a point code of 2-

1-1 and subsystem of 50. In this example, it is assumed that SSN 50 of SCP **410** provides CNAM-type database information.

Once again, when compared with the prior art network configuration illustrated in Figure 1, it will be appreciated that the inclusion of FEP routing node **500** in Figure 10 allows each of the provisioned SCPs **410**, **420**, and **430** to eliminate dedicated, internal FEP modules that would have otherwise have been associated with each back end database processor.

It will be appreciated that the network configuration shown in Figure 11 is perhaps the most powerful implementation of an FEP routing node of the present invention. It should be noted, as mentioned above, that in the example shown in Figure 11, all of the SCP nodes are identified by the same SS7 point code (2-1-1), which is identically the same point code assigned to the FEP routing node (2-1-1) yet each SCP node has a different subsystem provisioned for service. Once again, as each of the back end processors that comprise these SCP nodes is connected to the FEP routing node **500** via the IP data network **216**, each individual back end processor of each SCP will still have a unique assigned IP address in the form of a host name and a port. Those skilled in the art of telecommunications network operation will appreciate the implications and significance of such a network addressing scheme. From a practical standpoint, such a network architecture allows network operators to simply address all query messages to an FEP routing node, where the intelligence resides to determine which SCP or database should receive any given query message. Thus, as SCP or database nodes are added to the network, only routing information stored in the FEP routing node need be updated to reflect the network architecture change. In other words, the SCP or database portion of the communications network becomes essentially transparent to any service provider launching database queries. All the service providers need specify is the type of database service that is being requested (e.g., a subsystem or translation type), and the point code of the FEP routing node.

As such, when FEP routing node **500** receives a TCAP query message from SSP **110** that is destined for the SCP with point code of 2-1-1 and SSN of 50, FEP routing node might make a decision to route based on the operational status or congestion status of the multiple back end processors **432** that are associated with SCP **430**. Such load shedding or load sharing among multiple, similarly provisioned back end processors will ultimately lead to superior overall network performance. The FEP routing node **500** might further perform database protocol translation-type processing on the message, encapsulate the modified message in a properly addressed IP packet, and route the modified message on to the selected SCP back end processor. Once again, any additional processing that was previously performed by FEP **128** shown in Figure 1 could also be performed by the FEP routing node **500**.

It will be understood that various details of the invention may be changed without departing from the scope of the invention. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation--the invention being defined by the claims.